APRIL/88

ZX-Appeal

vancouver sinclair users group

next meeting:

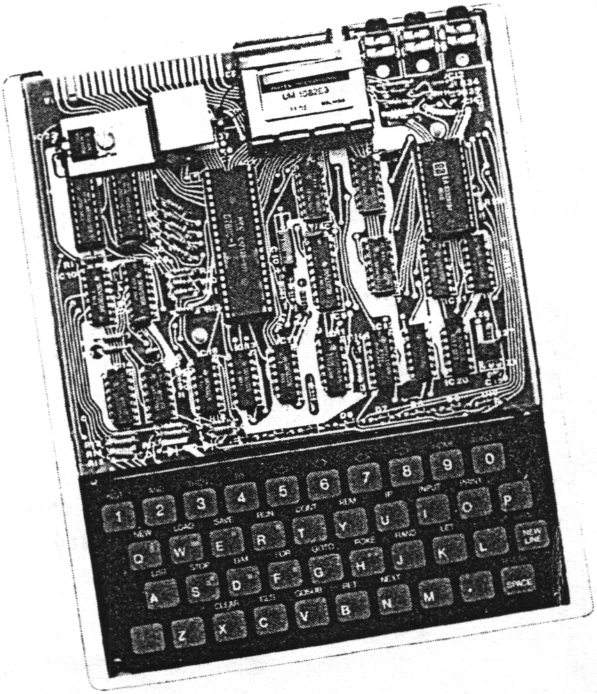
KILLARNY COMMUNITY CENTRE
6260 KILLARNY STREET
VANCOUVER

FRIDAY;7:00PM

April 8/88

ZXAppeal is a monthly newsletter put out by the Vancouver Sinclair Users Group. For more information on the group and ZXAppeal see the backcover.

inside



THIS ISSUE

Hey...what's going on? We were going to start alternating between 'fat' and 'skinny' issues because you guys couldn't come up with enough stuff to fill up the newsletter every month. (Also to make my task a little lighter every other month.) So what happens? I get so many submissions for this issue that it'll end up being 'fat' and I'll probably still not be able to fit everything in. Editors are never satisfied!

We start off with a 'New Prez Sez' from Gerd - so listen up. For those of you who don't know Gerd, we print herein his 'curricula vitae'. The ever innovative Vince L. has a 24k upgrade to the internal 8k NVM originally described in the Summer'86 Mega-Issue. Fred N. and Wilf R. are also never satisfied with leaving things alone - they sent along a nice article about improvements to Wilf's Hi-Res core routine that Fred uses in his Hi-Res programs. Reprinted is a letter received from VSUG'er Bill J. informing us of the formation of the Timex Sinclair Users Association of North America. Bill R. is back with Part II of 'Header Hacker'. The schematic for Tim S.'s On-Board Real-Time Clock for the 1000 is included this month. As well this month is another 'Playing with....' from Harvey. After all you QL'ers have built the modapter Harvey describes, you'll need to acquire a terminal program. Well it so happens that Meta Media has just what you need. Coincidence I'm sure. See their advert inside. Lastly, we reprint a letter received from a T/S user by the name of James T. Crumley. James's letter is self-explanatory and I'm sure he would appreciate receiving correspondence.

...meeting date!

APR/88

SUN	MON	TUE	WED	THU	FRI	SAT
*	*	*	*	*	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

March 11/88 Minutes

-by your humble scribe

Ken opened the meeting at 19:16 with 23 intrepid souls present; 5 others straggled in later. The first topic was kudos for Harry Slot on his impressive PC8300 article published in SyncWare News. Good show Harry! Ken then wondered out loud about the re-inking research which Harvey was allegedly doing. A show of hands gave evidence that many present had cloth ribbon type printers. The process continues.

Ken then related some interesting slander, namely a letter from an educational distributor of apple II-GS's, documenting various problems with the machines. The school board is not interested in Laser 128's apparently.

Then Ken read an interesting tidbit from SyncWare News regarding their publishing previously published material. It seems they want written clearance from the previous publisher. This item was enough to strike off yet another round of the olde copyrighte blues in ten part harmony, which somehow we managed to endure. A motion was passed that in ZXAPPEAL all rights revert to the authors immediately. Rod says he will add a line so stating to our newsletter banner.

Word has it that Fred Nachbaur is seriously considering designing a Timex clone. There was talk of colour, double width resolution & an "Oliger" board. God only knows what might become of this!

By the way, Fred N also has donated 3 more NVM programs to the ZX81 library.

Ian Mclean is continuing with his database project. He is wondering how this might be combined with a BBS, as well as wondering why more local TS'ers are not into modemming. [Drop into CityLink (604)222-2000, FrogHollow (604)469-0264 or NNN (604)354-4666 to catch the action.] Guido Vereira apparently has picked up an old Osborne with 2 disk drives & he is wondering about setting up a BBS with that. There was a lot of talk about different options here. Marcio Vereira was concerned about the cost of another phone line. There was discussion about a group purchase of the super cheap modem boards available now. Glenn Read piped up that he had some 1200 baud modems he would like to get rid of (real cheap). Suffice it to say that nothing was decided, but everybody is curious just where Ian's database project where lead him. He is agog at the possibilities. Later in the evening he was heard to ask Harvey if 'Hypertext' was explained in Computer Lib.

Ken took this occasion to break in on the conversation & thank everyone for letting him be prez & all; then dished out the same congrats for Rod & Harvey & absolutely everyone doing anything.

Rusty Townsend then announced that he had 12 QL raffle tickets left & he was holding the draw tonite. the meeting was broken up by a mad rush for tickets; several members were seen running down the hall for coffee.

Miraculously the meeeting came to order again in time to hear Rod Humphreys report that we have CAN$686.00 plus some last minute credits. The last balance was a bit high apparently because Rod had not paid the Dec/87->Feb/88 bills.

Ken, (the excumbent?), then raised the topic of the Bylaws & tried to interest us in passing / rejecting / amending said Bylaws. Marcio suggested that Items 8 & 9 ought to specify "members present at the meeting". A motion was so put & the Bylaws adopted.

Elections were held & the New slate is thus:

President: Gerd Breuning
Vice Pres: Glenn Read
Members at Large: Chung Chow, Guido Vereira, Vince Lee

There was general amazement at the quality & number of worthy candidates. A healthy sign for the club.

There was not much of a hardware report because Harry Slot is in Mexico [His wife won a trip for two.]

Ian Mclean, the ZX81 librarian, took another stab at his database wonderment by asking what information people wanted to have at hand. At present he is collating ZXAppeal articles/authors, a BBS list, Library books, ZX81 & 2068 progs. If you have any ideas, contact Ian.

Bill Rutter, the 2068 librarian, has apparently been in touch with George Chambers from Toronto & they have the same Cleveland tapes as do we. They find the tape quality to be similarly atrocious. Bill now has 43 tapes.

Rod then pointed out to Ken that Ken had skipped an item on the agenda, namely the editor's report. Ken was duly crestfallen & turned the floor over to Rod who announced he had nothing to say. (It's no wonder Ken felt persecuted.) Anyway, when the laughter died down, Rod pointed out the neat Julia Set Calendar which Fred N had sent him to include in the newsletter. He also plugged Dan Elliot again for his excellent TS repair work.

At this point Ken, the prez with seconds to go, asked if there was any other business from the floor. Ken Grant related an enchantingly bizarre tale of building a step up transformer for his 240v Philips monitor from an old TV transformer and some massive ceramic resistors he picked up cheap at RAE. He has the thing wrapped up in an industrial strength steel box complete with a recycled handle. There are door stops & there are door stops.

At this point Bob Denison took over with a demonstration tape of SAM, Software Automated Mouth, a text to speech system on the C64. The speech was intelligible, barely...

Kevin Kerney, the book librarian, had some doubles of magazines he was giving away. He is looking to "lighten his load".

Harvey spoke about receiving CTM magazine 2 years after trying to subscribe to TS Horizons. Then he spoke a bit about the new release of Computer Lib/Dream Machines (by Ted Nelson) & the idea of Hypertext. Harvey had also brought some QL related magazines & newsletters.

Eric Sakara piped up with the news that he had seen another article about the chap with the 7 computer bicycle, this time in '73' magazine.

Bob Denison made the QL Raffle draw & Dave Ross won.

Ken A. then related a twisted tale of a stray robin with a pungent bonnet, as going away advice to the new slate. The meeting slid generally into one on one conversations.

MESSAGE FROM YOUR NEW PRES.

After two years of inspiring leadership our executive has insisted on stepping down and letting others carry on. With the membership's, i.e. your participation we hope to see a continuation of the growth in enthusiasm and continent-wide membership which we were blessed with.

I would like to thank all members, including those from out-of-town, for your support and newsletter contributions. May I suggest that the most important feature in our "ZX APPEAL" is the one entitled "MINUTES-by your humble scribe"? - You see, it tells you what's happening during our club meetings. Whenever you can't participate in your club goings-on in person, maybe you could participate by sending or phoning in ahead of time what YOU would like to see done or discussed and I'll make sure it's placed on the agenda for the next meeting. This suggestion is of course not meant to be taken as a substitute for our EDITOR'S standing request "Keep those Newsletter Articles coming!" We eventually hope to have received from each one of you at least a Member's Profile. These are just a couple of suggestions how we could ALL get to know one another a litte bit more and expand communications.

Greetings to you all and lots of thanks to our outgoing EXECUTIVE for a job well done.

Gerd Breunung

****

PACIFIC COAST FAIR
TIME: SATURDAY APRIL 30,1988
10 AM TILL 4 PM.
PLACE: MCPHERSON CENTER
7325 MCPHERSON AVE. BURNABY
PRICE:GENERAL ADMISSION-
$2.00 TABLES-$25.00 BEFORE
APRIL 8 $35.00 AFTER APRIL 8.
PHONE:270-0064 FOR PRE-
REGISTRATION.
THERE WILL BE DOOR PRIZES,NEW
& USED EQUIPMENT,USER GROUPS,
AND REFRESHMENTS.

MEMBERSHIP PROFILE

It all started at the Robson Square Media Centre on Sat., Sept. 17, 1983. All the different user's groups were there with display tables and impressive demos. The most impressive to me was the Timex-Sinclair table: there was Ken Abramson's computer in a see-through case so you could see the inside, Karl Brown's machine was controlling things with an add-on board which he had designed and built himself, and John Brohman and Bob Lussier told me enthusiastically about all the wonderful things you can do with this little computer. Best of all, they told me I could pick one up today complete with a 16 K Rampack for the incredibly low price of only $ 99.99 at A&B Sound. I rushed out, made it to the shop just before closing time and joined the club!

Since then I have added a 2040 Printer, a 2050 Modem (thanks again, Harry Slot, for making it work and the beautiful case), an external Keyboard, an Admate TP-80 full size thermal Printer with Tasman Centronics Interface, and some spare machines and rampacks (solicited from people who had moved on to "better things"). My son and I are having hours of fun with the extensive software which I collected over the years: games, educational, home and business. I have completed all of Karl Brown's projects: Parallel I/O Board, Sound Board, Speech Board, as well as BERT (his robot) which I program with my T/S 1000 using a home built RS 232 Interface. Also, I have built almost all of Wilf Rigter's projects: Joystick Interface, Speech Board, Built-in 8 K NVM and three 32 K NVM Boards.

Currently I have a subscription for Syncware News. I would like to do some control projects and learn more about machine code and FOURTH as soon as time permits. It will take years for me to explore all the posibilities and capabilities of my T/S 1000 system. This realization has prevented me from aquiring one of Sinclair's more advanced computers, or any other. I am a draftsman by profession and I am waiting for an AUTOCAD version to come along that will run on my T/S 1000! ...

Gerd Breunung

PLAYING WITH ELECTRICITY

- Mar 26/88
- Harvey Taylor

What the hell is a modaptor anyway?

One of my greatest disappointments with the QL when I originally got it was the serial ports. When I discovered I couldn't just hang a garden variety modem off the serial ports, I was po'ed to say the least. I have written in the past about the reasons for these problems, however in brief the serial input for both ports is multiplexed onto one serial input on the 8749 IPC [Intelligient Peripheral Controller] which has to watch the keyboard & make noise & keep time as well. Consequently the serial port has a tendency to drop characters now & then.

Another consequence of this arrangement is that the programmer does not have direct access to the device which controls the serial parameters. ie Data bits [5-8], Stop Bits [1-2], & Parity [none,even,odd,mark,space]. You can ask QDOS to ask the 8749 to change the Parity & Baudrate, however the internal ROM of the 8749 is forever (?) hidden from the programmer. If you're driving a printer, this doesn't much matter. However if you want to get input from a modem in North America, problems arise. One problem is that just about every BBS in the world expects you to use 1 Stop Bit. The QL expects you to use 2 Stop Bits. The Modaptor is a hardware device that gives you two stop bits. It also serves the function of generating a 75 Baud output signal from a 1200 baud output signal. I have not looked at this function because it is only used in the UK with Prestel & those systems which expect the 1200 baud receive/75 baud send protocol.

I have heard of people getting modaptors with the part numbers defaced so that reverse engineering is impossible. For some reason mine came readable. This is the circuit.

These are the other connections between the QL & the modem

QL [SER2]	*CONNECTIONS*	*MODEM [DB-25]*
1 - GND	------------------------------	*7 - GND*
2 - TxD	------------------------------	*2 - TD*
3 - RxD	------------------------------	*3 - RD*
4 - nc		
5 - nc		
7 - Tied to QL Pin 1		
9 - +12v	-----------------------------	*4 - CTS*
9 - +12v	-----------------------------	*20- DTR*

Note that the QL User Guide in Concepts [Communications RS-232C Page 13] refers to the British Telephone connectors used in Europe, not the DB9 connectors used in NorthAm. The DB9 pins 6,7,8 are tied to frame ground.

To say that I was somewhat puzzled upon first unwrapping this circuit would be putting it mildly. Here is what I think is happening. The 555 Timer is being used in monostable multivibrator mode.

Most of the time, the input is high, ie. the modem is marking. Remember the serial line is inverted. The high input keeps the transistor turned on, effectively grounding capacitor CØ. This keeps the trigger input of the 555 low, as well, which ensures that the output [pin 3] is high.

When a start bit comes along, ie the input drops to low, the transistor turns off and the capacitor CØ begins to charge via R2 and D1. When the trigger level passes a "certain critical voltage", the 555 output will drop low. When the output goes low it will pull the serial output low for a time period determined by R5,R6,R7,R8, the switch setting and capacitor C1.

When the input switches high again at the end of the serial data, the capacitor CØ loses it charge via R3,R4 and the switch setting. This hold time has the effect of adding a second stop bit for the QL, I think.

It is possible to make some calculations to check these conjectures. There are three equations in particular. The charge time of CØ, the discharge time of CØ and the pulse width according to R5-R8 & C1. The rise time of a circuit is the time required to go from 1Ø% to 9Ø% of the input value. The rise time of the circuit consisting of R2 and CØ is described by:

```
     Tr = 2.2 RC
        = 2.2 * 47ØØ * 3.3E-9
        = 3.41 E-5 seconds & this is fast enough to be instantaneous.
```

The fall time of the CØ & R3/R4 circuit is described by:

```
     Tf = 2.2 RC
        = 2.2 * (24ØK + 68K) * 3.3E-9
        = 2.23E-3 sec.
```

or if the switch is on,

```
     Tf = 2.2 RC
        = 2.2 * 68K * 3.3E-9
        = 4.9E-4 sec.
```

The pulse width of the 555 Timer is described by:

```
     Tw = 1.1 * Rt * C                    where Rt= Total Resistance
        = 1.1 * (R5+R6+R7+R8) * 24E-9
        = 1.1 * (1.Ø7M) * 24E-9
        = .Ø282 sec.
```

or if the switch is on,

```
     Tw = 1.1 * Rt * C
        = 1.1 * (R5+R6) * 24E-9
        = 1.1 * (267K) * 24E-9
        = .ØØ7Ø5 sec.
```

Consider the Serial data stream.

```
1  ----+    +----+----+----+----+----+----+----+----+----+----+----+----+
    M  / S  / 1  / 2  / 3  / 4  / 5  / 6  / 7  / 8  / P  / St   St   M
Ø      +----+----+----+----+----+----+----+----+----+----+
       / Tc /                                                      /
       /<---------------------- Tchar ---------------------------->/
```

HEADER HACKER (PART II) by Bill Rutter

Early Spectrum software had only rudimentary protection built in, but as piracy became a serious problem, these methods became exceedingly sophisticated. While it can be said that no method offers complete protection, the time required to solve the problems may be more than the pirate is prepared to spend.

Programmers have many ingenious ways of preventing software piracy, and these make it very difficult to make back-up copies, and cause problems for hackers who wish to examine and/or modify programs.

The best way to deal with this is by using an interface like Multiface, which lets one stop, copy, examine and modify the contents of any register at any time. Lacking this, the first task is to try and break into the Basic Loader by MERGING, or using the short routine from Part I. If the header lists in Basic, the appropriate POKES can be inserted before the final PRINT USR or RANDOMISE USR statement. If the screen is blank with the o.k. message, try changing the PAPER or INK color, as the INK and PAPER may be the same. If this fails try LIST I, or LLIST to the printer. If the first line is 0 (zero), try

```
   POKE 1+PEEK 23635+256*PEEK 23636,1
```

to make the first line #1, and thus editable. If all this fails, and the first line is 0 REM, orjust 0,then the loader is in machine code, and cannot be dealt with without a disassembler.

In the case of Adventure Games where text is inputted, it is sometimes possible to LIST or LLIST the entire vocabulary that the program will recognise. Either MERGE "" the loader, and note how many blocks of CODE load after the SCREEN$, or LOAD"" and observe how many blocks of CODE seem to load. First load the code ONLY, then try the following -

```
10 FOR x=24201 TO 55000: IF PEEK x<VAL"65"
OR PEEK x>VAL"122" THEN NEXT x
20 PRINT CHR$ PEEK x;: NEXT x
```

Don't be surprised to see some rude words occasionally (e.g. GREMLINS)! This works for all the KET TRILOGY, SPIDERMAN, EUREKA GREMLINS, et al.

Part III will review and explain some of the more common methods of programme protection.

The Abbreviations stand for:

M	Marking	High
S	Start bit	Low
1-8	Data bits	Either
P	Parity bit	Either
St	Stop bit	High

Note: There is no difference between the Stop Bits & when the modem is Marking.

On a 300 baud modem, there are 300 of these transitions per second. ie Tc = 1/300 = 3.33 milliseconds

Depending on the number of data bits & etc, Tchar is 7-12 times as long. The common NorthAm standard of 8,N,1 is 10 slots long, so Tchar=33.3 msec.

On a 1200 baud modem, the misnomer of 1200 baud arises to ensnare the unwary. A 1200 "baud" modem uses a 600 baud signal, but backs twice as much information into a transition by changing the encoding. At any rate, for a 1200 "baud" modem:

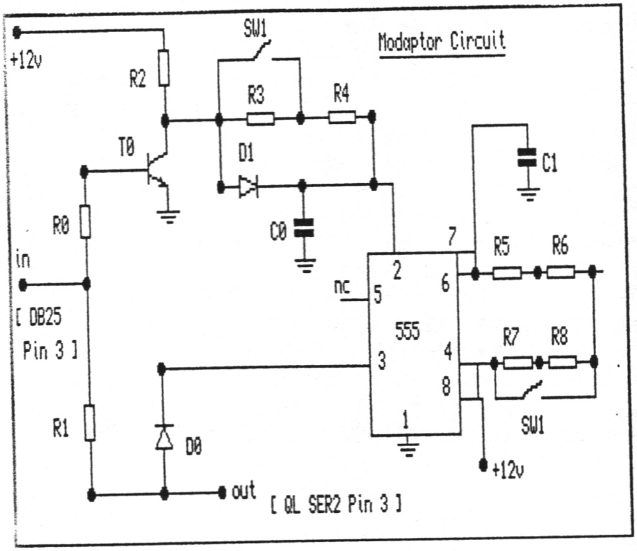
Tc = 1/600 = 1.67 msec.

And for the 8,N,1 standard, Tchar = 16.7 msec

Now is the time for caveats. I have not built one of these circuits myself, although I intend to eventually. Normally I would not publish information of which I was not absolutely certain, however I know there are a lot of people who either refuse to pay US$50.00, or can't afford it. Consequently I am releasing this, let us say, as Stage One of an ongoing project.

I would like to know any experiences you may have playing with this circuit. I will pass along any useful information in further articles. If you have any comments or corrections, please pass them along via the editor.

eof>



PARTS LIST

D0	1N418	
D1	1N418	
C0	3n3	(3.3 nanofarads non-polarized poly)
C1	24n	(24 nanofarads non-polarized poly)
T0	BC182A (NPN)	
R0	100K	
R1	4K7	(4.7K OHMS)
R2	4K7	
R3	240K	
R4	68K	
R5	240K	
R6	27K	
R7	470K	
R8	330K	
IC	555 Timer	
SW1	Double pole (ie. both switch at the same time)	

NOTE: All resistors are 1% metal film

This article will appear in a future issue of SYNCWARE NEWS.

WRx16 REVISITED!

W. Rigter with F. Nachbaur

Since our publication of Wilf Rigter's WRx16 high-res discovery (SWN Vol. 4, #2), several enterprising folks have been hard at work, developing new refinements in high-res and other "cheap video". Mr. Rigter worked up a new version of the "core," overcoming some previous hardware limitations. Gregory C. Harder, W. C. McGrath, and undoubtedly many others have made other discoveries. I worked up a couple "Interesting, maybe even useful" routines presented in this article.

But whoa! our esteemed Editor (and probably most other folks) thinks that this is going to be yet another of Fred's incomprehensible but presumed sensible high-tech ravings. Not at all. This is a "play by doing" thing that let s you do neat things with those little TS1000's or TS1500's.

There WILL be theory sections, for your information later on, once you feel the urge to go beyond basic experimentation. The possibilities are many and varied, but there are some inevitable limitations. The theory sections will help you learn to more fully control your machine's video.

For now, though, skip these sections and just enter, enjoy, and play with the demos.

HISTORY

To make a long story short, high resolution has always been possible on the ZX81/TS1000. It has also, somehow, always seemed out of reach, ignored, or both. Until WRx16.

In 1986, Wilf Rigter demonstrated a system of high resolution display to the Vancouver Sinclair User Group, one of the most if not THE most active ZX81 hardware group in existence. The ZX computer had just a little tack-on board, and it produced high-res screens. No internal mods were needed.

It later resulted in the commercial high-res version of SINC-ARTIST (Callisto Software) dubbed THRUST by its distributor, Weymil Corp. Because of the large number of "Hunter" board owners, the program found its way to many users, and became a classic "must-have" for this machine. It represents the entry-level of high-quality graphics development, giving you an Artist program with many features of programs for larger machines.

The rest is "history." When we ran it over a year ago, we had no idea that this would stir as much of a resurgence of interest in the ZX81. One short 79-byte program and a small memory board, and it makes this computer once again a viable tool in a world of IBMs, Ataris and others. The saga continues....

DESCRIPTION

WRx16 is a system of high-resolution (256x192) display using simple multi-purpose hardware; a suitable non-volatile 8K static RAM, such as the one we ran in SWN 4:1. What's more, programming for it is comparatively simple. An Extended BASIC is available, and several applications have sprung up using this system. There are even programmers working with this in the U.K. It should be interesing to see what they come up with.

Here and now, there are yet more vistas in display exploration for the ZX81-family. Some will allow the creation of a windowing 128-line Editor routine, which allows instant horizontal scrolling of the 64-column window. New possibilities for modemming exist, involving the use of a 1200-baud modem for the machine. All at very low cost.

On that subject, the VSUG newsletter, ZX-Appeal, reports in the Feb-Mar '88 issue that TS1500's with TS2040 printer are available for as low as $30 from Electronic Surplus Inc. in Cleveland. This, plus a small static memory board, is a small investment for a high-res-capable learning system. You can even do "real work" with it, as communicate to BBSs and info services. In the future, you will be able to do far more.

HARDWARE

What do we mean by "suitable static memory?" This is the only thing that might be construed as a "catch." Fortunately, you have several options.

1: Modified Hunter board. See the Theory section for the mod. It just involves soldering two small, cheap parts, and cutting one trace of the board. This popular device was sold into the thousands years back, and is still a popular commodity to haggle over with other users.

2: Built-in NVM. See SWN 4:1, for a circuit worked up by Wilf for Gerd Breunung, who developed it into a truly professional construction article. This just plugs into the 2K RAM socket on the board, with seven small wires to connect to the circuit board. No trace-cuts are necessary, if you are concerned about that but aren't afraid to do a little "kitting."

NOTE: If you haven't already built and debugged this little project, here's some more motivation for you. We're working up a way of easily adding another 32K of static memory. By adding another chip, this same tiny board can now be a real powerhouse!

3: Tim Stoddard's 64K RAM. This was published in Time Designs Magazine and (in abbreviated form) in Computer Shopper. This is a neat modification to your ZX81/TS1000 that fills the full 64K with static memory. It requires several trace cuts and jumpers, so may not be up everyone's alley. TDM also ran Tim's article on interfacing the LarKen DOS to the high-res system, which should be applicable to any RAM arrangement.

An important note: All developments to date assume static RAM in the entire 8-16K region. If you wish to use other things that use this memory, you will need a way to disable either conflicting device.

4: Commercial alternatives. I sell an update of the Hunter board, using 1-8K I.C. instead of 4-2K chips. Being more cost-effective, I can sell it for less than the price at which the Hunter board sold. However, I ask that you allow me at least several weeks for delivery, since parts availability seems to be a chronic problem.

A better bargain for many users will be Wilf Rigter's own 32K NVRAM. He also worked up a very good RAMDOS for it, allowing you to keep an entire collection of short programs (or a couple long ones) for instant availability when you power up. It is easily re-mappable, using a series of DIP switches, to result in virtually unlimited options with other hardware. You can even enable RAM in the ROM region, and modify the operating system to your heart's content. Note, the character sets cannot be modified.

MOVING ALONG

So you have the basic picture. Why is this worth doing? Have someone at your local group demo this for you. You might just be amazed. There's already a lot of great software commercially available or in public domain. There are entire collections of interesting pictures available. You will step into the world of "real computing" with a small inexpensive communications device.

Yes, communications. I didn't say "computer" because that can be a scary word. [You REALLY have to be wary of people like me, who often refer to "the machine" in articles.] You can't use a computer if you cannot see what's going on, or otherwise communicate with it. WRx16 improves your vision by a factor of 16. The resolution is just as good as the TS2068, in normal display mode, except without the color. The quality of the display much better; it is clear and rock-steady on any reasonably good monitor. For people who, like me, don't have a NEED for color in a "serious" computer, its lack is actually an advantage because of the resulting simplicity.

In a future installment we'll tackle the last remaining gripe against the machine; its speed. The result will be a machine (preferably TS1500) running almost twice as fast as before (in SLOW mode), with full software compatability. All at a cost of only a few dollars. Add a couple more bucks for a switch, and you can use either mode. There are even software-only methods, such as "Quick" mode in SWN 5:2. At this writing we are investigating even far more significant possibilities for speed (and possible resolution) increase.

LET THE FUN BEGIN

In this first article of a possible series, we present a new WRx16 core. This overcomes several (virtually all) limitations of the original routine published in SWN 4:2. This is a GOOD THING TO DO to your existing WRx16 software, since it makes it usable with a greater range of hardware combinations. New programs should use this routine to prevent problems later.

I'll also demonstrate a couple interesting alternatives, using demos that you'll find in the listings.

The goal is to make you want to understand it all better, and before long you'll be poring over the theory section and your ROM disassembly, etc. Don't be afraid to experiment. You will then be on the fore-front of a wave of development that may yet result in a very inexpensive yet competent personal communications computer. This wave has already seen working prototypes of easy-to-use 256K bank-switching. It promises new hardware advances in display resolution. It can be made operable at low power, making it an interesting and useful tool, midway between the lap-top and the desk-top. It combines some advantages of both, at a fraction of the cost of either.

Best of all, (I've been saving this little tidbit all this time), your machine will be virtually always be repairable. We're working on ways of building such "Super-ZXs" onto a single board, containing NO custom chips! (If you don't believe that it's possible, look at a ZX80 schematic!) So even if your Timex ULA or SCLD goes out, your machine can still live on. You can afford to devote interest, time and money into a device that can greatly enhance your personal productivity. You don't have to wonder if your computer will still work next week. Now isn't that a pleasant thought in trying times?

WRx16 VERSION 2

The WRx16 core that we published in 4:2 works fine with the usual 16K RAMpack on a ZX81/TS1000. It also works, with two restrictions, on most 64K packs. The first is that the dynamic RAM should preferably be switchable in the 8-16K region. This restriction still exists on the new version; we can't do much about that. However, even some RAMs that can't be switched out here, will work in spite of the apparent conflict.

The other restriction is that we had to take special measures in software to ensure that a portion of the high-res system be copied in the top 16K block. This can get messy, since it will corrupt any other data you might have up there.

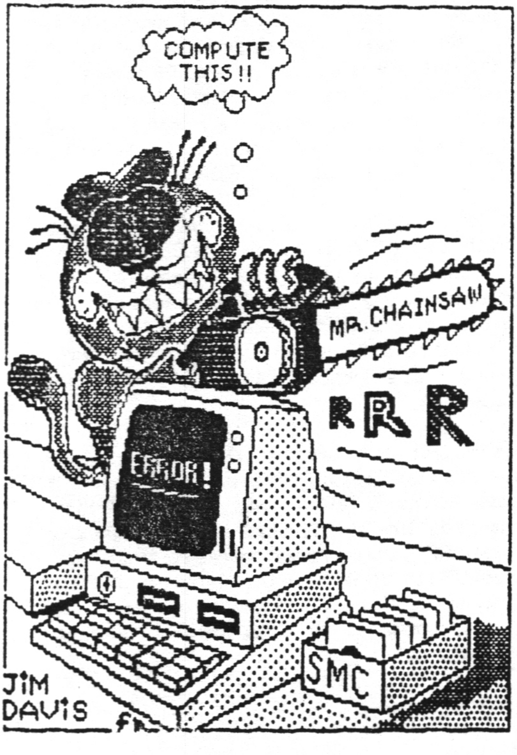
WRx16 Version 1 does not work at all with certain hardware combinations; e.g. the TS1500 alone (without external 16K pack), and the Memotech 32K pack alone. Both will work if a standard Timex 16K or similar pack is added. Refer to the Theory and Listing 1 (disassembly) to understand why this is.

The new routine, refined by Wilf Rigter, took a few more bytes than the original routine. I (fn) then did a little byte-pynching to make it fit into the same space as before. It can therefore replace existing software, such as the utilities and demo given with the original article.

ENTERING THE NEW ROUTINE

If you entered the version 1 demo, your task is much simplified. Use the loader (GOTO 9000) to enter the values of Table 2. Use START=16514, BYTES=128 just as when filling the 1 REM line in the original article. If you prefer, use Hot Z and Listing 1 instead. After saving the new version to tape in case of bugs, run the demo. It should work as before. The difference is that it will work with virtually any memory configuration. (Including 2K, incidentaly, provided the necessary 8K static RAM is present.)

If you're new to this, follow the article in 4:2, replacing Table 1 with the version presented here.



THEORY

WRx16 V1 will not work if the 48-64K block is fully decoded. This is because it relies on the high-memory echo of the dummy display file. If this is not present, either in "echo" RAM or in real RAM (as 64K), the routine crashes. With 64K, we can physically place a copy exactly 32768 bytes higher than the "actual" dummy display file in the 16-32K region. However, with either of the 32K configurations mentioned above, the 48-64K region will be truly blank; no RAM, and no echo of the 16-32K block either.

More precisely, only the last two bytes of the DDF need exist in high memory. To understand why, let's review how the line-scanning system works. On receiving an interrupt, the main display loop is started. After taking care of housekeeping (counting lines), the program jumps to the high-memory echo of the dummy display file. When an M1 cycle (instruction fetch) occurs with A15 high (original ZX81) or A15 and A14 high (TS1500 or ZX81 with "Oliger mod"), the ULA takes over control. It uses the (previously set) I and R regiters as a pointer to the address (in low memory, i.e. bit 15 is held low) being displayed. This is what causes the hardware to read the contents of memory directly during the refresh interval provided for dynamic RAMs. This is why we have to AND REFSH* and RD* when controlling the SRAM's chip select. (On the Hunter board, for instance, cut the trace from RD* to CE*, bridge the gap with 4.7K, add diode from CE* to REFSH*, cathode towards REFSH*.)

The final result on the data lines is then fed serially to the TV. Meanwhile, the CPU gets fed NOPs (00h) instead of the true contents of the instruction being "fetched." It therefore delays for 4 t-states, exactly the time it takes to display 8 horizontal dots. (Have you ever wondered why the clock crystal's frequency is twice the CPU clock frequency?)

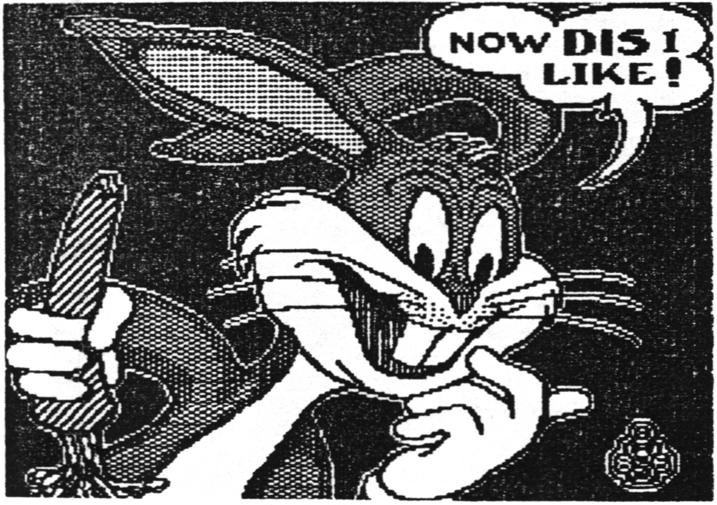
This process continues until all 32 horizontal bytes have been displayed. At this point, the code of the instruction being fetched will have bit 6 high. (Commands like HALT, used in the normal display system; RET, used in quasi hi-res routines, and JP as in WRx16 V1 all have this in common.) This is a signal to the ULA that display is complete, and it returns control to the CPU. The CPU then executes the command.

The problem with JP is that during the data-fetch part of the cycle, when the address to which to jump is being obtained, the CPU will still think it is running in high memory (PC has bit 15 high). If the RAM is fully decoded, (i.e. no "echo" or duplicate of the low-memory copy) it will only find garbage there, and jump to some uncharted never-never-land.

Version 2 fixes this by using JP (IX) to return the line-scanning operation to the main program, instead of a JP to a fix address. This way, there is no data fetch after the instruction fetch, and the high-memory echo is no longer requird. (Note also that JP (IX) meets the requirement of having bit 6 of the opcode set.)

To compensate for the difference in timing that results from using this approach, some of the "delay" commands had to be changed. This moves some of the entry points, and is why we are presenting the new routine in its entirety. The timing is about the only thing that could mess you up in experimenting. It is very critical, and has to reach certain key points at exactly the right T-state. So if you change anything, you have to be careful that the timing of your new code matches that of the old. You do this by trimming the dummy timing commands. These are marked in the listing as DELY, and timing loops are marked TILO. Get out your Zaks handbook or other Z80 reference, and keep track of T-states.

Finding suitable timing values with a minimum of required space, while not interfering with your program, can be quite an interesting challenge. Beware of some, like RRD, which appear to be great, compact time-consumers, but may not do what you expect (RRD and RLD operate on (HL) as well as A).



ENTERING THE PROGRAM

THE EASY WAY

One way to do this one is from the ground up. However, if you have followed the demos so far, you can save some time. Load the demo program of 4:2, and delete the demo portion (everything between and including lines 7 through 540). Leave line 5 in place. Enter line 4,

```
4 REM BIPLOT
```

Perform the following POKEs:

```
  POKE 16645,4
  POKE 16646,1
  POKE 16419,4
```

Then delete line 5. This will "melt together" the REM lines 2 and 3, giving us enough room for the modified service routines. It also keeps us from getting stuck in the long REM lines at the beginning. Henceforth, you may replace line 4, but not delete it. If you do you'll have to pull the plug to regain control. If you ever get in trouble after LIST, and the listing is stuck, LIST 4 and then POKE 16419,4. That'll fix it.

FROM THE GROUND UP

If you start from scratch, enter a 1 REM line exactly 128 characters long. This is four full lines on the screen, looking like this:

```
  1 REM xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxx
```

Enter another REM line, numbered 2, long enough to hold at least our present operating system. To leave some elbow-room for any of your additions later, make it, say, 512 bytes long. (FAST mode will make this somewhat quicker.) Then enter the loader program of Listing 2. RUN this to input the values of Table 2 into the 1 REM, and the values from table 3 into the 2 REM.

THE BASIC

Now enter the BASIC program of listing 4. After saving with RUN 9990, the program will auto-run.

Have fun with this. As a final note, let us know if you want to see more reports of new developments. Some possible topics that already exist include: Using SHR-Extended BASIC from machine-code, to end up with a smooth, believable hi-res implementation of the classic SPACE INVADERS game. Using "QUICK" mode with WRx16, or with quasi-hi-res programs. PRINTing characters in high-res.

So once again it's up to you; what do you want us to share with you?

THEORY

Listing 3 shows the disassembly of the modified core and service routines. To make the dual display work, we first have to make some changes to the WRx16 core. From 40BAh to 40C5, we have added a new routine which checks bit 0 of FRAMES. The subsequent commands cause HL to be loaded with 2000h (HRDF #1) or 3000h (HRDF #2), depending on the state of this bit.

When viewing only one display file, the command at 40BD is changed from AND 01 to either LD A,00 or LD A,01. These commands take the same time to execute, so this was just an easy way to insure that the loop will run the same regardless of whether we've viewing just one, or both of the display files.

Note that the command at 40AD was changed, and a NOP was added at 40C5 to compensate for the extra time it takes to run our new code. If you count T-states, you'll see that the overall timing remains exactly the same.

Note also that we load B with only 80h at 40B2, since we are only displaying 128 hi-res lines.

The only other changes are at 40E3 and 40ED. These determine how many low-res rows are displayed. Since the hi-res portion takes 16 rows, we can display 8 lo-res rows for the same 24-row overall display size. However, one row is lost in returning from our hi-res system; this is why we have to kill some time at the start of DP-3, to wait for the start of the first line in the next available row.

The value loaded into DE at 40E3 represents 8000h plus the offset from the start of the normal display file, to the first byte of the starting low-res row to be displayed. The command at 40ED loads B with one more than the number of low-res rows, and C with 7 (always).

Just as our service routine modifies the command at 40BD to change viewing modes, the commands at 40E3 and 40ED could be changed to give a low-res "windowing" option, allowing you to scroll the entire low-res display up and down in the 7-row low-res portion of the display. Use a SYNC routine similar to ours here, to insure that both changes are made right after a vertical sync pulse, since you want to be sure that both changes will be in place by the start of the next field.

The M/C and BASIC program listings will be in next month.

THE AMAZING BI-PLOT

We all know, of course, that the daffy-nition of BI-PLOT is "the purchasing of a devious plan." Or any number of other bizarre things, which I'll leave you to decipher.

Ah, but let's get back to the real world. This article gives a demo of a way to do high-resolution grey-scale graphics. That's right! Though I may be stretching the definition, we'll actually see the ZX81/TS1000 display a high-res scale of three shades - black, white and grey. A quote I just invented runs, "Three shades doth verily a grey-scale maketh."

We'll once again use bizarre methods to put that remarkable ZX hardware/software combination to work, deviously planning to make it do things even that IT never thought possible. Best of all, you won't have to purchase a thing (assuming that you're with us so far on this high-res stuff).

For the sake of the demo, let's go back in time. Way back.... do you remember SWN Vol. 1, kids? This contained a lot of (sometimes embarrassing) "early Fred Nachbaur," including a "GreyPlot demo" that I also used in my CE-AMP program (now in shareware). Well, what could be better than to repeat that early demo, using this modification of WRX16?

WHAT IT DOES

BIPLOT works much like "QUICK" mode (SWN 5:2) does, in the sense that it uses the built-in FRAMES counter (FRMS in Hot-Z-ese) to display different things in alternate fields of the TV display. Unlike QUICK, though, we have a continuous display, even though the speed is no faster than normal SLOW mode. (The good news is, we can use FAST mode to do all our plotting, then get into compute-and-display mode to view and play with the result.)

The way we do it, is to toggle between two different high-res display files. Anything plotted in both display files shows up as black (or white in inverse mode). Anything plotted in only one display file will only be displayed every other time. As a result, these points will look "grey." Anything else is in the background color (white or black, in normal and inverse modes respectively).

In the demo, two sets of axes are drawn, one in each display file. However, different horizontal axes are printed just for demo purposes. Two different functions are plotted into each of the display files; their product is printed in both.

After the display has been created, you can view either hi-res display file, or both. You can also reverse the video at any time.

THE DEMO

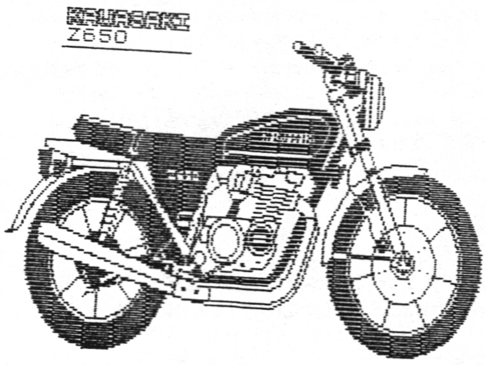
This demo displays several of WRX16's unique features:

1: Bit-mapped video file. This makes our plotting, etc. extremely compact (the whole mini-operating system takes less than 256 bytes). It's memory-mapped, allowing us write to it from BASIC, using POKE.

2: Any or all vertical columns can be reversed. To reverse a column, change the corresponding element of the dummy display file to 128 (decimal). Our "reverse" (RVRS) routine simply toggles all 32 locations (columns) between 0 and 128 whenever it is called.

3: WRx16 is extremely adaptable. We can change the number of hi-res and low-res lines at will. We can do different things in different fields of the TV display. We can even do "QUICK" mode! (Topic for a future installment, if the interest is there.)

4: Since a full hi-res bit map of a 256*192 screen takes 6K, and we only have 8K available, we have to cut down the size of the hi-res portion to 2/3 of its full size. This gives two 4K blocks that can be used for two display files. With the high-res display files in high memory (as with Tim Stoddard's setup or with Delta), these could possibly be brought back to full size. However, this version demonstrates yet another neat trait of WRx16: you can display any combination of hi-res/low-res rows on one screen.



The 24K Internal NVM

By V. Lee

Back in 1986, the July edition of ZX Appeal and the September edition of Syncware News published an article on how to install 8K of non-volatile memory inside a TS1000. It allowed us to have the benefits of the Hunter Board to store not only machine code utilities, but Basic programs as well. With 32K NVM now available, new software have been written to take full advantage of this additional memory. Let's upgrade our 8K to our 24K NVM. Designed to work with the TS1016 Rampack, this new memory sits in the 8-16 and 32-48K region.

This entire circuit is built right inside the computer using 28 gauge ribbon cable. A soldering iron with a ground is required. And ground yourself by touching the regulator heatsink before handling the ICs. The keyboard ribbon will present a problem. Extending it with standard ribbon cable may allow it be connected. I decided to leave it disconnected and run an external keyboard. Note that all the signals from the keyboard diodes are taken from the cathode side. And MREQ, WR, and ROMCS signals are taken from the rear card edge connector. Look for a TS1000 that has its internal Ram chip socketed. Otherwise a socket will have to be installed. A ZX81 with a 28 pin socket can also be used, but changed the jumper beside pin 21 from L1 to L2. This switches the signal from VCC to A10.

Begin by installing the 74LS10. This chip decodes the M1 line to allow machine code to run in the 32-48K region. The pins are flatted out and the chip sit backwards on top of the Z80 CPU. This allows easier access to the signals. Remove the Z80 from the board when you're soldering to it. A Reset switch is also thrown in. Remove the ULA chip from its socket and bend pin 10 outward and connect it to pin 8 of the 74LS10. If you've never added a heatsink to the ULA chip, I would recommend purchasing one to increase the life for this IC. 40 pin IC heatsinks are available through various electronic outlets. Make all the necessary connections, install the chips, plug in the Rampack and apply the power. If the inverse K doesn't appear, disconnect the power and recheck all your connections.

Next, install the 74LS138. This decoder chip acts as the ON/OFF switch for the 6264. For test purposes, connect pin 3 of the 74LS138 to a 330 ohm resistor to ground.

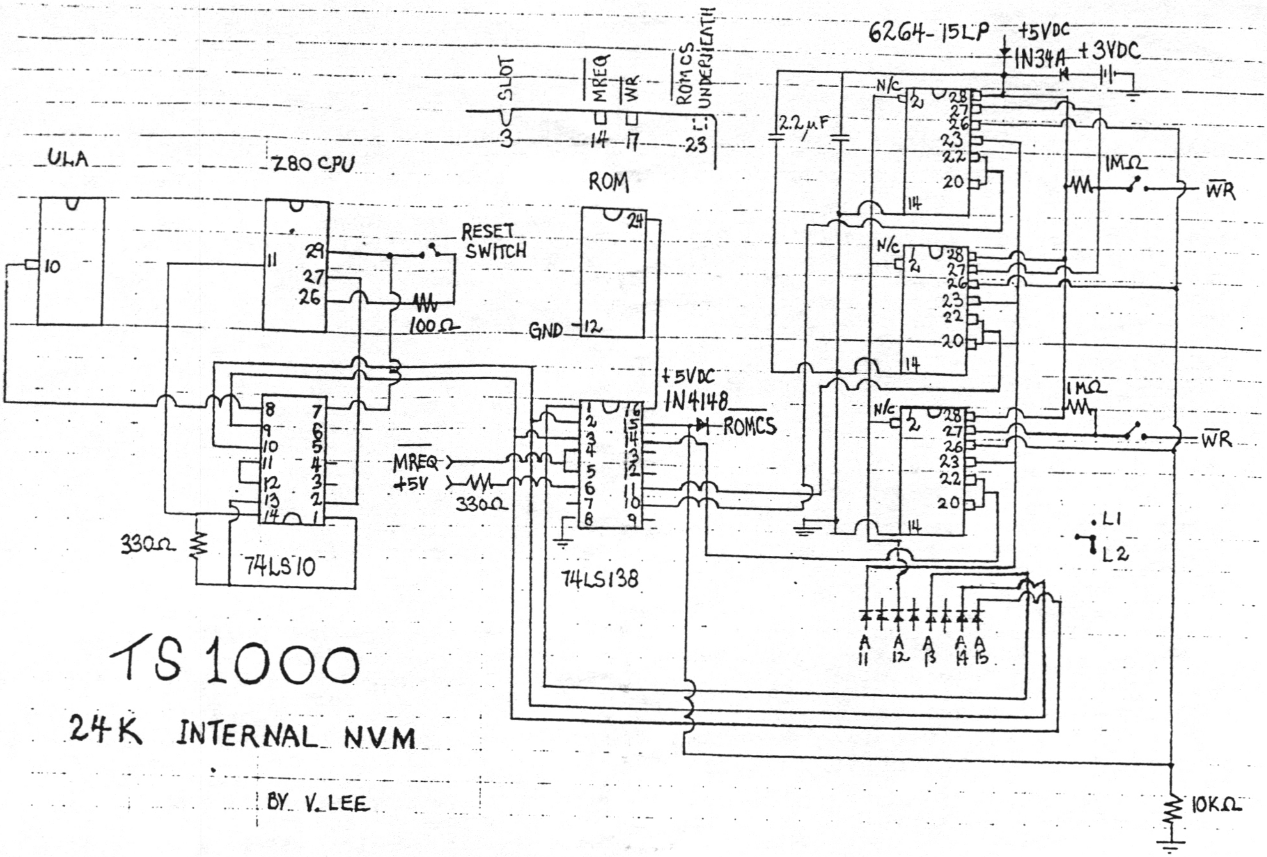
Install only one 6264 at this time. Pin 2 is bent out and connected to the third keyboard diode while pin 23 is bent out and connected to the first diode. Pin 27 is bent out and connected to the Write Protect switch. Pin 28 is bent out and is connected through a diode to VCC and through a diode to 3VDC. Pin 1 is bent out and is not connected. Tie pin 20 to pin 22 and connect them to pin 13 on the 74LS138. Connect the negative end of the battery to pin 14 on the 6264. This pin, along with the rest of the remaining pins, are connected to the socket. Close the Write protect switch and apply the power. Do not plug in the Rampack. 8K of memory has been placed in the 16-24K area. If the inverse K dosen't show up, remove the power and recheck all the connections. If you peek into location 16389 you should get 96. Test the other two ram chips with this procedure.

Disconnect the 330 ohm resistor and connect pin 3 of the 74LS138 to the last keyboard diode. Disconnet the ram chip and stack the three ram chips on top of each other. All the pins are joined to their respective pins except for pins 20, 22 and 27. Each pins 20 and 22 on the 6264 are tie together and joined to their proper pins on the 74LS138. Pin 27s are connected to their proper Write protect switches. Since this is the final assembly, use care in the placement of the wiring, especially behind the 6264. The screw for the casing runs through this area. The Reset switch, the Write protect switches, and the battery holder are all mounted on the outside.

Once completed, install the Rampack and plug in the power. Close the Write protect switch, poke 192 into location 16389 and execute NEW. Ramtop should now be set to 48K. Use the routine from Vol 4#1 to check out the ram in the 8-16K region.

You now have a 24K Hunter Board built right inside a TS1000 for storing programs and utilities. Or, if you like, reset Ramtop and you have a 32K system for storing variables. You're also ready for Hi-res software from Silicon Mtn. using WRX16 core routine. Achieve an incredible display.

For those who crave for more memory, next time we'll look at the 32KX8 static ram chips for the 96K NVM system.



TS 1000

24K INTERNAL NVM

BY V. LEE

QTY	PARTS LIST
3	6264LP-15 STATIC RAM CHIPS
1	74LS138 DECODER I.C.
1	74LS10 NAND GATE I.C.
2	1N34A GERMANIUM DIODE
1	1N4148 SWITCHING DIODE
2	2.2UF TANTALUM CAPACITOR
2	1 MEG 1/4W RESISTOR
3	330 OHM 1/4W RESISTOR
1	100 OHM 1/4W RESISTOR
1	10K OHM 1/4W RESISTOR
1	PUSH BUTON SWITCH N/O
2	TOGGLE SWITCH SPST
1	3V BATTERY HOLDER

Mr. James T. Crumley
Drawer "A"
Fort Leavenworth, KS 66027-7140

March 23, 1988

Vancouver Sinclair Users Group
Rod Humpreys, Editor
2006 Highview Place
Port Moody, B.C. V3H 1N5 Canada

Dear Mr. Humpreys,

I am an incarcerated Timex/Sinclair 2068 user looking for other users to correspond with. Due to limitations imposed by the institutional administration however, I am unable to join outside clubs or groups. Couple this with the fact that I do not, at the present time, have the necessary funds to join such organizations and it makes for a terrible situation for someone trying to learn more about computers.

While I have quite an extensive theoretical background in computer science (I hold a masters degree in computer studies), I have very little actual programming experience due to my current circumstances. I would like to correspond with persons who are knowledgeable in the various languages, other than BASIC, that can be utilized on the T/S 2068. Especially the Hawg Wild implementation of fig-FORTH. I have a copy of this last and cannot seem to get the colon definitions to compile from the tape as the instructions (which are quite sparse) say it should.

I have plenty of time to answer letters and would really like to start a serious correspondence with people who are interested in the Timex/Sinclair computers. I look forward to hearing from you soon.

Sincerely,

James T. Crumley

TS·2068 UP·DATE
the user's NEWS

Take it to the limit!
A quarterly Magazine for the users

TS-2068 UP-DATE
1317 STRATFORD AVE.
PANAMA CITY, FL 32404

March 10, 1988

TO:

George Chambers   Ian Robertson   Bob Mitchell
Ken Abramson   Rod Humphreys

I am just back from the Orlando Winter Fest where all TS-2068 and QL products were selling like hot cakes. Even this old cheap skate bought a QL plus about $400.00 of accessories. Now gotta enter the learning curve again, durn it! Due to the vigor and numbers of the QL users, and the fact that about 1/3 of the UP-DATE subscribers also have the QL, the JULY issue of UP-DATE will begin a QL section. Hopefully some good writers will help.

The biggest news at the Fest: The Florida group came prepared to instigate a "National Timex Sinclair Users Association". I quickly put in my 2 cents and proclaimed that it should be a "North American" association, and the fact was cited that the Toronto (Ontario), and the Vancouver user groups are two of the most active and strong Zx-TS groups in the World. Now you guys and gals must back me up with your participation!

The Association idea was received with great enthusiasm and a working group was established for organization and planning. Tim Woods (Time Designs magazine) and I both volenteered editorial and communications space to publicize the Association. Other key persons will be announced, but as of now, Mary Lynn and Eric Johnson, 249 N. Harden Ave, Orange City, FL 32763, are the contacts. Also, Eric's BBS will be the information and coordinating center, tel- 904-775-0093- 300 baud 8-1-none. You may have trouble getting this BBS until a pestering hacker quits antagonizing.

Things are looking up for the TIMEX SINCLAIR user group with the influx of 5-8000 new QL users and the HUGE Z88 group that is about to hatch. I hope that you will stir up your Canadian users to be supportive of a North American Timex Sinclair User's Association and participate in the planning and organization- even volenteer for staff positions. I am sure that your participation will be appreciated. Best regards to all.

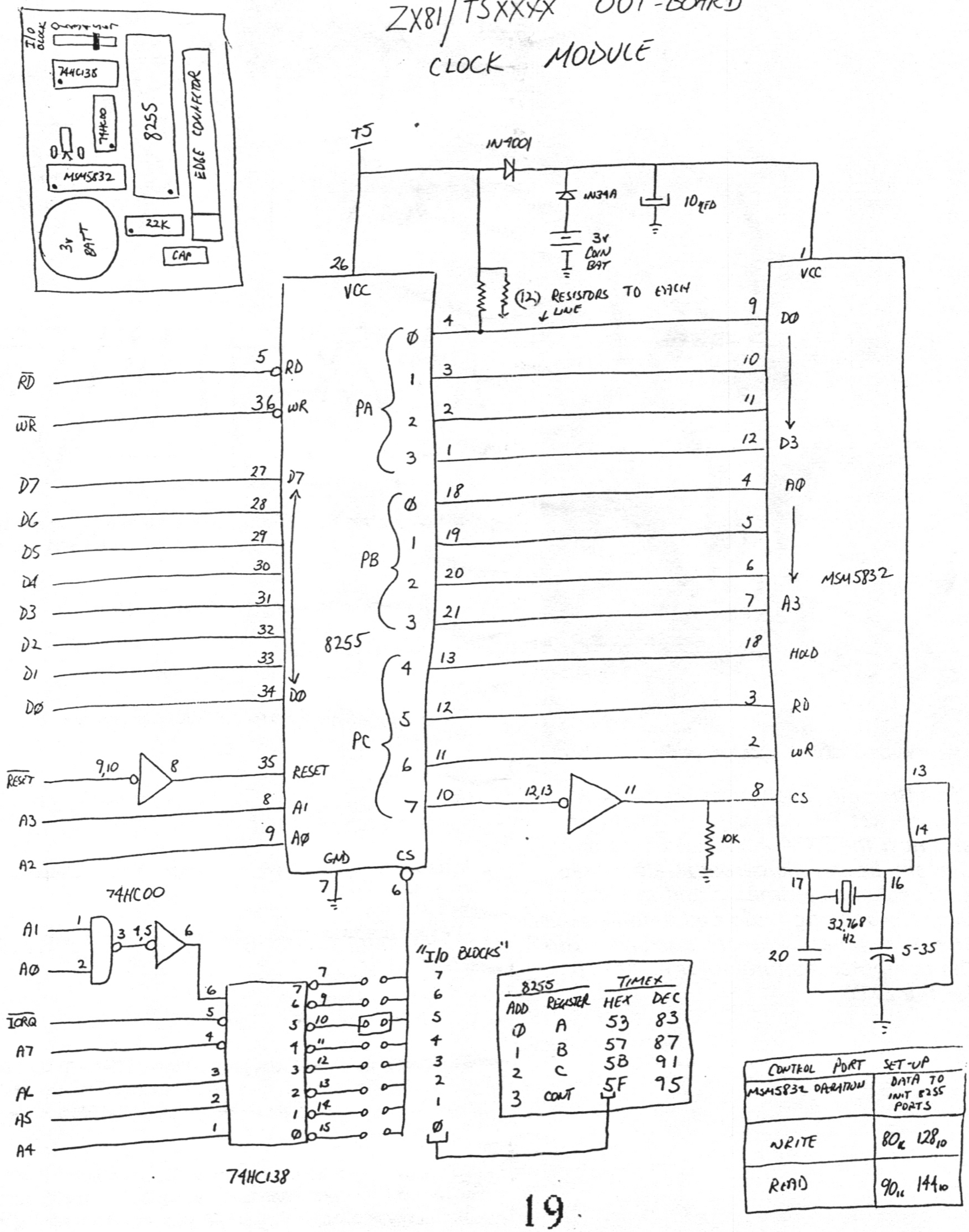
Sincerely,

Bill Jones

Bill Jones

This is the schematic for Tim's article in last month.

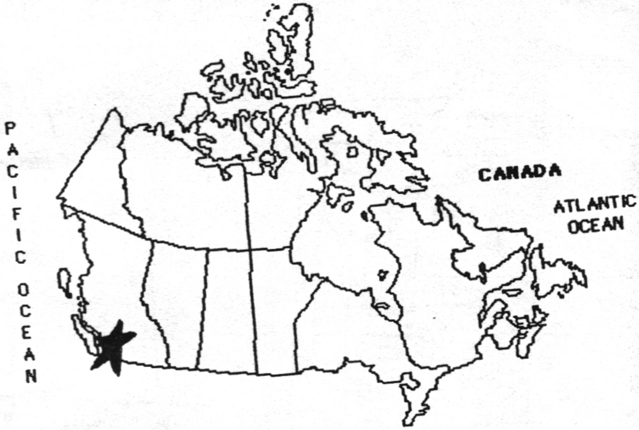
ZX81/TSXXXX OUT-BOARD CLOCK MODULE



8255 ADD	REGISTER	TIMEX HEX	DEC
Ø	A	53	83
1	B	57	87
2	C	5B	91
3	CONT	5F	95

CONTROL PORT SET-UP	
MSM5832 OPERATION	DATA TO INIT 8255 PORTS
WRITE	80_{16} 128_{10}
READ	90_{16} 144_{10}

Clive Sinclair, inventor of the world's first pocket TV, the pocket calculator, and now the ZX80 Personal Computer System, began his career in electronics as a technical journalist and in 1962 began producing radio and amplifier kits. His company has now evolved into the computer manufacturer responsible for the ZX80.



VSUG

The Vancouver Sinclair Users Group has been in existence since 1982. We are a support group for the owners and users of all SINCLAIR and TIMEX computers.

Pres:- Gerd Breunung PH*(604) 931-5509
V/Pres:- Glenn Read
Sec:- Harvey Taylor
Treas. & N/L Editor:- Rod Humphreys

Our membership dues are only $15.00/year and may be sent to the Treasurer:

Rod Humphreys
2006 Highview Place
Port Moody, B.C., V3H 1N5

Members of VSUG receive a monthly issue of ZXAppeal - our newsletter.

ZXAppeal accepts advertising. Our **PREPAID** rates are:

$10.00	full page
$8.00	1/2 page
$5.00	1/4 page

ZXAppeal is distributed to approx 30 other SINCLAIR User Groups throughout North America as well as overseas via the NETWORK. NETWORK correspondence may be directed to the Editor at the above address.